Evaluation of Alternative Arbitration Policies for Myrinet Switches*

P.J. García, M.D. Mora,
F.J. Alfaro, J.L. Sánchez

Dept. de Informática
Escuela Politécnica Superior
Universidad de Castilla-La Mancha
02071- Albacete, Spain
{pgarcia, mdmora, falfaro, jsanchez}@info-ab.uclm.es

J. Flich

Dept. de Informática de
Sistemas y Computadores
Universidad Politécnica de Valencia
46071- Valencia, Spain
jflich@gap.upv.es

Abstract

Interconnection networks consist of a set of switches interconnected by point-to-point links, and hosts linked to those switches through a network interface card. These networks are becoming increasingly popular as a cost-effective alternative to parallel computers. One of the most well-known example is the Myrinet network. These networks use techniques and incorporate components that have been successfully applied to interconnection networks for parallel computers. One of these techniques is the use of crossbar chips as the main component of switches, allowing to connect the input channel of any of their ports to the output channel of any of their ports for forwarding incoming packets. Of course, when several packets request the same output channel, certain criterion must be used for selecting one of them. This criterion may affect performance in terms of latency and throughput, specially in wormhole networks (Myrinet) where a packet occupies several resources while it is waiting for the output channel. In Myrinet, an arbitration policy based on round-robin criterion is used. However, there exist other arbitration policies that, when applied to Myrinet networks, could increase their performance. In this paper, several arbitration policies for selecting the packet for a free output channel have been considered and evaluated. We have found that strategies based on injection time and remaining distance to travel reduce the average and maximum packet latency of the current policy used in Myrinet.

Keywords: Myrinet, NOWs, Arbitration, Performance

*This work was partly supported by the Spanish CICYT under Grant TIC2000-1151-C07 and by a scholarship of the Junta de Comunidades de Castilla-La Mancha.

1. Introduction

One of the main characteristics of modern high-performance interconnection networks is the use of different switching techniques. Unlike traditional medium-shared local area networks, the hosts of a switch-based network send and receive messages by means of point-to-point (usually full-duplex) links connected to switch ports. The switches of the network, on their hand, can connect the input channel of any of their ports to the output channel of any of their ports for forwarding incoming packets. Packets may cross several links and switches until they reach their destination. Because of switches are multiport components, switch-based networks support many concurrent communication paths, achieving high throughput.

Obviously, the optimization of the functions performed by switches is a key objective on the design of switch-based networks. These functions vary depending on the techniques used for switching and routing packets. In networks with distributed routing, like ServerNet [6, 4], switches perform three main functions [2]: routing, selection, and arbitration. The routing function supplies a set of output channels for each packet being routed. The selection function selects a free output channel (if any) among the set supplied by the routing function. Arbitration is needed when several packets simultaneously request the same output channel in order to decide which packet will be forwarded.

In networks that use source routing, like Myrinet [1], the routing function is really performed by the source host because it determines every output channel of all switches along the route for each packet. Each switch must only take the packet and forward it to the output port indicated by its header. No selection function is needed because the routing function supplies only one output port. Therefore, the arbi-

Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'02)
1530-2075/02 $17.00 © 2002 IEEE

tration function is the only one used at Myrinet switches in order to take any kind of decision. Due to this fact, network performance could be greatly influenced by the arbitration policy used. To the best of our knowledge, Myrinet arbitration function is based on a round-robin scheme. However, other arbitration policies [5, 9] can be applied to Myrinet networks. Therefore, it would be interesting to evaluate the impact of different possible policies on Myrinet network performance. In this paper we present a simulation-based comparative evaluation of several arbitration policies based on different criteria and applied in a Myrinet-like network.

The rest of the paper is organized as follows. In Section 2 we review some aspects of Myrinet, specially the way switching and routing are implemented. In Section 3 we define the arbitration policies that have been evaluated. The simulation methodology and results are presented in Section 4. Finally, in Section 5, some conclusions are given.

2. Myrinet

Myrinet [1] is a Gigabit-per-second switch-based network developed by Myricom Inc. [7]. The main characteristics of Myrinet are the use of source routing, wormhole switching, centralized reconfiguration and stop & go flow control on every link. Myrinet is mainly used as a Local-Area-Network in clusters of computers or workstations with irregular topology.

In Myrinet, the transfer of messages between the host and the network is controlled by the Myrinet Control Program (MCP). The MCP is executed by a RISC processor included on every Myrinet interface card. The MCP also provides other network management functions, including mapping and monitoring the network.

Myrinet performance can be improved by using application programming interfaces (APIs). Myrinet software is open for customers, so it can be easily customized. Some research groups have developed their own Myrinet control programs and APIs, achieving very small host-to-host latencies and high throughput. For instance, BIP [10] (Basic Interface for Parallelism) is a message-passing system implemented on top of Myrinet that achieves one Gigabit/s bandwidth and less than 5 μs latency (with 1.2 Gbps Myrinet fabric).

2.1. Myrinet Switching and Routing

Myrinet packets are source-routed: every packet in the network follows a path fixed by its source host. Usually, hosts use the up*/down* routing algorithm to compute deadlock-free routes from them to any other host in the network. Given the destination of a packet, the source host inserts in the header of the packet information about every port to be used along the route to the destination host. Myrinet switches read the first byte of the header of an incoming packet to determine the requested output channel. Next, this byte is removed and the CRC of the message is recalculated. The next switch of the route will read and remove the new first byte, and so on until the packet reaches its destination host. Therefore, the initial header must include a byte for each switch of the route, and the header size will be decremented by one in each switch. Because routes will have different lengths (that is, will include a different number of switches), the initial size of the header will be variable.

On the other hand, Myrinet uses wormhole switching for forwarding packets through the network. When this switching technique is used, switches can forward a packet before it has been completely received. So, Myrinet switches read the first byte of a packet, determine the output channel and forward the packet to this channel before the packet tail has arrived. If the requested channel is free, this process takes 150 ns approximately. Moreover, wormhole switching allows fragments of the same packet (flits) to be buffered in several switches of the route. Because of this fact, wormhole switching tends to reduce the buffer size requirements. In Myrinet, buffers are associated with input channels, and their size varies from 25 to 100 bytes depending on the type of interface.

Of course, if the output channel requested by a packet is not free, the packet is blocked and must be buffered instead of being discarded. Because wormhole switching is used, the packet blocking will affect several switches depending of the packet size. In Myrinet, a blocked packet can remain partialy or completely in a buffer until it is forwarded (the requested output channel becomes available) or a timeout expires. This timeout is on the order of 4M character periods (aproximately 25 msec.). Timedout packets are dropped to avoid destination blocking, source blocking, and misrouted packets that could lead to deadlock.

The operations that Myrinet switches must perform are so simple that are hardware-implemented. Myrinet switches are based on crossbar-switch chips and Myrinet-interface chips. Both types of chips are implemented in VLSI technology. The number of crossbars and interface chips composing the switch depends on the number of switch ports. There are Myrinet switches with 4, 8, and 16 ports.

3. Arbitration Policies

At a given time, there may exist several packets buffered in a switch requesting the same output channel, and therefore some criterion is required for scheduling the forwarding of these packets. The arbitration policy defines this criterion, and so it is applied when any output channel is freed to select a new packet among the ones that are waiting for being forwarded to this channel.

The arbitration policy that Myrinet implements is defined as "recirculating token", that is, is based on a round-robin criterion [1]. In this case, the selection of the packet to be forwarded is done in a cyclic way among the buffers containing a packet. However, there is variety of packet or switch characteristics the arbitration policy could be based on [5, 9]. Some of these are: the length of the packet, the contention accumulated by the packet along its path, the location of the source node, the location of the current node, the location of the destination node, the distance already travelled by the packet, the distance to travel still, the time the packet was sent, the time the packet arrived at the current switch or the input channel chosen previously at the current switch.

We planned to analyze the behavior of Myrinet switches considering some of the arbitration policies mentioned above: random, injection time, waiting time [5], distance to travel still, distance already travelled and round-robin. The most relevant aspects of these policies are explained in the following items:

- *Random*: A message is selected in a random way. It is the simplest policy, but it has a serious drawback: the time a message is waiting for its output channel is not bounded.

- *Injection time*: The oldest packet is selected. This policy attempts to avoid high latencies. As Myrinet is an asynchronous network there is not a global clock in order that we can compare injection time of packets from different source nodes. This policy could be implemented if we add a few bits in the packet header and each switch adds the time the packet takes to output it and the time spent travelling over the links.

- *Waiting time*: The packet that has been waiting for a longer time at the current switch is selected. In this case no more information must be included in the packet header, but some bits should be added at each port to compute the time a packet has been waiting.

- *Distance to travel still* (DTS): The packet with the least remaining distance to reach its destination is selected. Myrinet headers have already this information: the distance to travel still is the number of remaining "routing bytes" of the header. However, a certain logic must be included in the switches in order to compare the information of the arrived packets.

- *Distance travelled* (DT): The packet which has completed more hops of its route will be selected. In Myrinet headers, this information is not included. So, a few bits should be included in the packet header in order to keep track of the distance currently travelled. At each intermediate switch this new field should be increased.

- *Round-robin*: The selection is done in a cyclic way among the channels containing a packet waiting to output for the channel freed. This is the current arbitration policy used in Myrinet.

An advantage of the random and round-robin policies is that they do not need additional information nor additional hardware. On the other hand, policies based on injection time and distance travelled require that a new field should be added in Myrinet packet headers. In this sense, requirements for distance to travel still policy are much minor, because this information is already in the packet header. However, a new logic should be added in order to obtain this information.

To sum up, all the new policies proposed (except random and round-robin policies) require to add new hardware to Myrinet switches in order that the comparisons and additions could be done. These new comparisons will take time. However, these tasks can be done in parallel with other tasks (i.e. obtaining the output channel for the packet) and even while the previous packet is leaving the switch. Therefore, we assume these new tasks will be out of the critical path of packets.

4. Performance Evaluation

In this section, we evaluate the behavior of the arbitration policies. We have used a flit-level simulator that models the Myrinet network. First, we will detail the network and the traffic patterns we have used, and the parameters used in the simulations concerning links and switches. Later, we will present the obtained results.

4.1. Network Model

The network is composed of a set of switches and hosts, all of them interconnected by links. Network topology is completely irregular and has been randomly generated taking into account some restrictions: there are exactly 4 hosts attached to each switch, all the switches have the same size (8 ports), and two neighboring switches are connected by a single link. These assumptions are quite realistic and have already been considered in other studies [12, 13, 3]. Regarding network size, we have evaluated networks with 8, 16, 32, and 64 switches, and, for all cases, the results obtained exhibited a similar behavior.

Regarding the routing algorithm, we have used the well known up*/down* routing algorithm.

4.2. Link and Switch Models

To model the links, we assumed short LAN cables to interconnect switches and workstations. These cables are

10 meter long, offer a bandwidth of 160 MB/s, and have a delay of 4.92 ns/m (1.5 ns/ft). Flits are one byte wide. Physical links are also one flit wide. Transmission of data across channel is pipelined [11]. Hence, a new flit can be injected into the physical channel every 6.25 ns and there will be a maximum of 8 flits on the link at a given time.

Virtual channels have not been used since current Myrinet switches do not support them. A hardware stop & go flow control protocol [1] is used to prevent buffer overflows. In this protocol, the receiving switch transmits a stop (go) control flit when its input buffer fills over (empties below) 56 bytes (40 bytes) of its capacity. The slack buffer size in our Myrinet simulator is fixed at 80 bytes.

To model the switch, we assumed that each switch has a simple routing control unit that removes the first flit of the packet header and uses it to select the output channel. A crossbar inside the switch allows multiple packets to traverse it simultaneously without interference.

Each output port can process only one packet header at a time. When a packet gets the routing control unit but it cannot be routed because the requested output channel is busy, it must wait in the input buffer until its next turn. At this point, we have considered that when the output channel is freed, and there are several packets that request it, certain criterion must be used for selecting one packet. In particular, the six strategies detailed in Section 3 will be evaluated.

4.3. Traffic Patterns

Message traffic pattern is greatly dependent on application. We have considered some of the most commonly used message destination distributions: Uniform, with locality, hot-spot, and bit-reversal. The uniform traffic pattern sends each message to any of the other hosts with equal probability. For the message distribution with locality, the destination host is, at most, l switches away from the source host, and is randomly computed. Two values of l have been considered: $l = 3$ and $l = 5$. Due to space limitation, we will only show results for a maximum distance of $l = 3$. The rest of results are available in [8].

In the case of hot-spot distribution, a percentage of traffic is sent to one host. The selected host is randomly chosen. The percentage of messages sent to the hot-spot has been set to 20%. The destinations for the rest of the traffic are randomly generated using a uniform distribution. Finally, for the bit-reversal distribution, the destination of a message is computed by reversing the bits of the source host identification number.

As Myrinet networks allow any packet size, we also have used different packet sizes. In particular, short and long packets have been considered: packets with 32 bytes (20 %) corresponding to control traffic, and packets with 2048 bytes (80 %) corresponding to application traffic.

For each simulation run, we have considered that the packet generation rate is constant and the same for all the hosts. Each simulation was run until the network reached a steady state, that is, until a further increase in simulated network cycles did not change the measured results appreciably. Once the network has reached the steady state, the flit generation rate is equal to the flit reception rate.

4.4. Simulation Results

In this section, we show the evaluation results. We will present results corresponding to the most important performance measures: average message latency[1] and throughput[2] We have also considered the maximum message latency. In general, the policies that allow us to obtain the best network behavior must have the smallest value for this measure.

In Figure 1, the average network message latency is shown for a network with 16 switches and 64 hosts. Results are presented for the four message destination distributions considered. In this Figure, we can observe that the results are similar in all cases. It can be seen that the use of the DTS policy leads to the best network behavior. Note that this strategy selects the packet with the least remaining distance to reach its destination. In most cases, the use of the DT policy leads to the second best behavior. This alternative selects the packet which has completed most hops on its route. The use of the policy based on the injection time (which selects the oldest message) also obtains better results than the round-robin policy. Note that all these strategies have in common the aim of prioritizing the packets seizing more resources. Thus, the sooner these packets are selected, the sooner resources are available and, therefore, lower network contention is encountered.

As we can see, regarding network throughput, the DTS policy increases throughput by a factor of 1.3 with respect to the original Myrinet policy (round-robin) with uniform distribution (Figure 1.a). Similar increases in throughput are also obtained for other traffic distributions (Figures 1.b, 1.c, and 1.d).

Regarding latency, we can observe that DTS significantly reduces the average latency. The DTS policy is the one with the lowest latency for all evaluated points of traffic. For medium traffic, average latency is decreased by roughly 40% for the uniform traffic distribution.

The same tendency in the results can be observed in the group of plots shown in Figure 2. This Figure presents results corresponding to the throughput per switch versus traffic per host. In this Figure, the policies giving priority to

[1] Latency is the required time to deliver a message. Latency is measured in ns.

[2] Throughput is usually defined as the maximum traffic accepted by the network, where traffic is the flit reception rate. It is measured in flits/ns/switch.

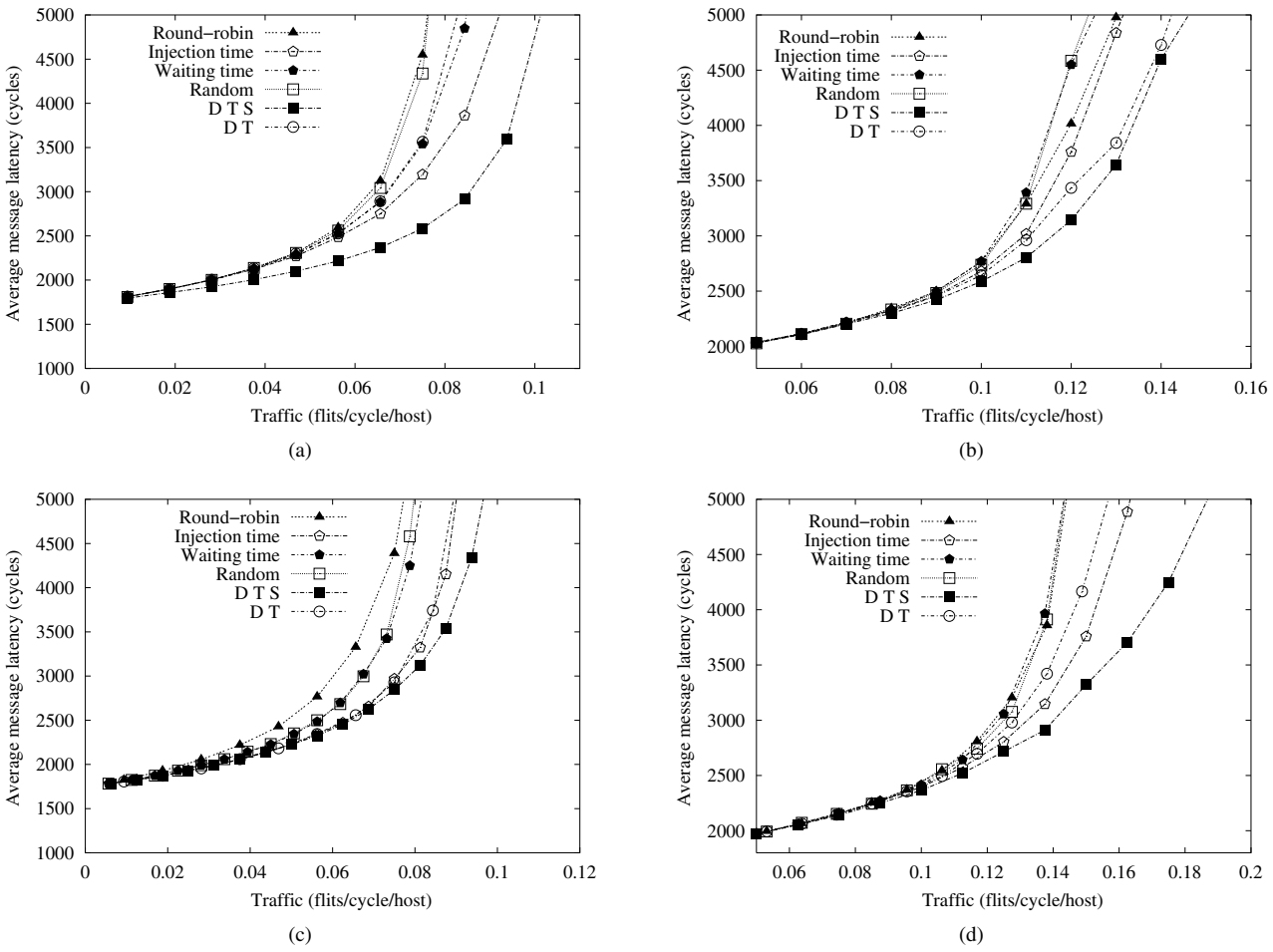

Figure 1. Average message latency versus traffic for the destination distributions: (a) uniform, (b) bit-reversal, (c) hot-spot, and (d) local traffic ($l = 3$). Network size is 16 switches and 64 hosts.

packets that use a greater number of resources (DTS, DT, and injection time) also allow the network to accept more traffic beyond the saturation point for the other policies.

In Figure 3, the maximum message latency is shown. This latency is presented for three traffic loads: low, medium, and high loads. The traffic corresponding to high load is close to saturation. Note that traffic loads are different for each message destination distribution. When the network handles a low load, all the policies have a similar behavior. In general, for a low load there is almost no contention in the network and, therefore, when a packet arrives at a switch, the requested output port will be available.

On the other hand, for higher loads, the differences are significant. In this case, it can be observed that the use of the DTS and DT strategies also achieve a good network behavior in terms of maximum latency. However, the use of the policy based on the injection time always leads to the best results. Although all of these policies have in common the aim of prioritizing the packets that, in general, have been waiting for a longer period of time in the network, it is more suitable to use the injection time criterion (due to its more accurate estimation of time) in order to reduce the maximum packet latency. This argument is confirmed by the results obtained. Note also that the round-robin policy (the one used in Myrinet switches) always has a worse behavior than DTS and DT. These results also confirm that starvation is not introduced in the network with the new policies.

Finally, in Figure 4 we have shown, for different network sizes, the average message latency versus traffic using the hot-spot message destination distribution. In all cases, the same behavior can be observed: the use of the DTS policy allows to obtain the best network performance. The use of DT and injection time policies lead also to good results.

To sum up, we have seen that the round-robin arbitration policy used in Myrinet can obtain a low performance when compared with more efficient arbitration policies. In

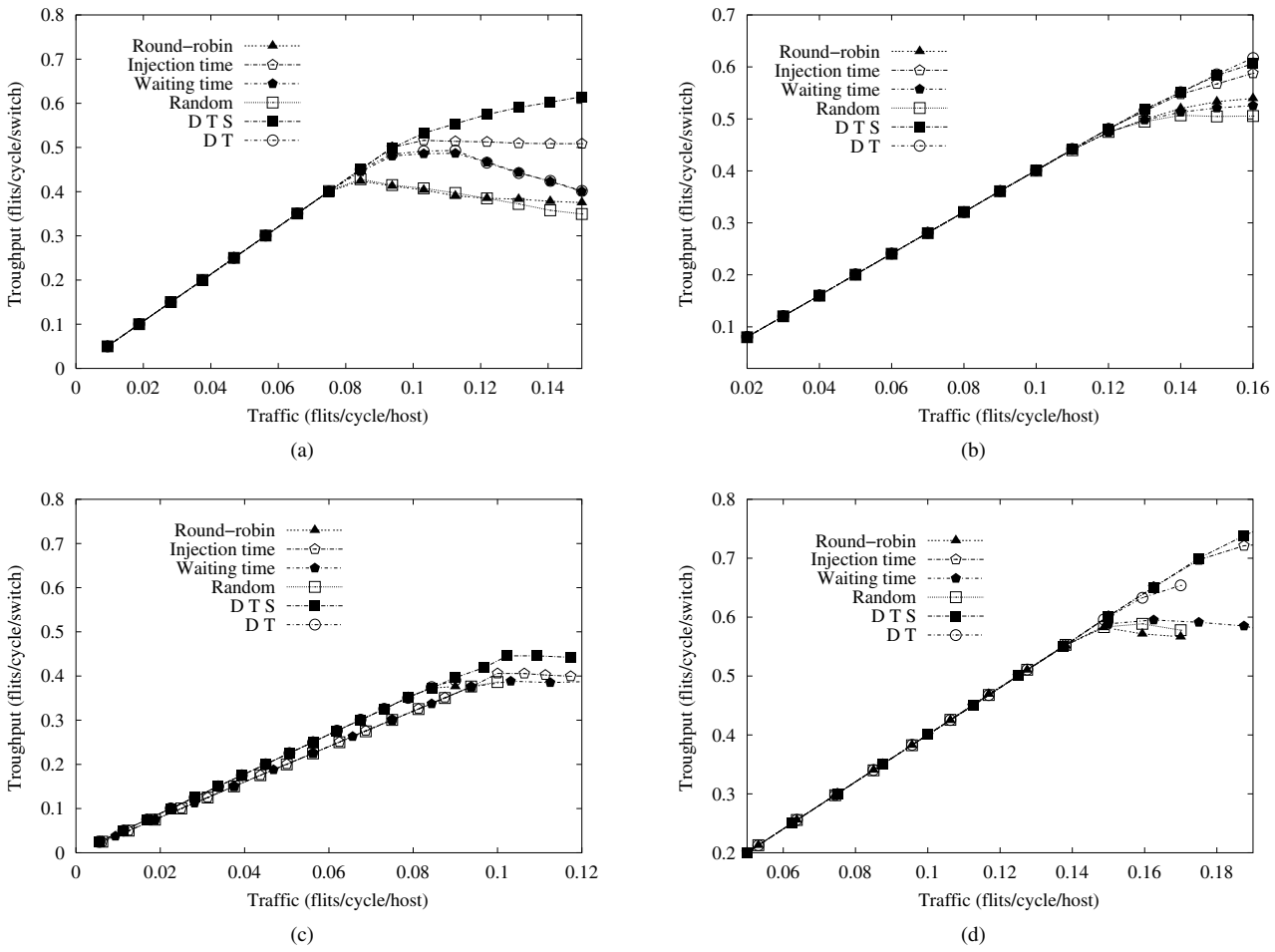

Figure 2. Throughput versus traffic for the destination distributions: (a) uniform, (b) bit-reversal, (c) hot-spot, and (d) local traffic ($l = 3$). Network size is 16 switches and 64 hosts.

particular, those policies that give higher priorities to those packets that are more time in the network. The DTS and DT policies increase network throughput and decrease average and maximum packet latencies, by handling more efficiently packets in high traffic loads.

5. Conclusions

Nowadays, the amount and the quality of network resources demanded by applications grow in such a way that any possible method oriented towards improving the performance of the interconnection networks that support these applications must be considered and deserves to be studied. The main objective of our work has been to test the network behavior using arbitration policies different than the one currently implemented in Myrinet, in the hope that some of these new policies might improve the network performance. Five alternative arbitration policies, described in Section 3, have been added to the Myrinet simulator used in our experiments. From the results displayed in Section 4, we can extract interesting conclusions.

For low traffic, independently of the network size and messages destination distribution, the behavior of the network is similar whatever arbitration policy is used. Logically, this is due to the fact that the lower the traffic, the lower the network contention.

For higher traffic, and mainly near saturation, the behavior of the network changes depending on the arbitration policy used. We have found that the use of at least three new policies (Distance Travelled, Distance to Travel Still and Injection Time) improves significantly the performance results obtained when using the current Myrinet round-robin policy. This improvement varies depending on the network size and messages destination distribution. However, average improvements are in terms of increasing network

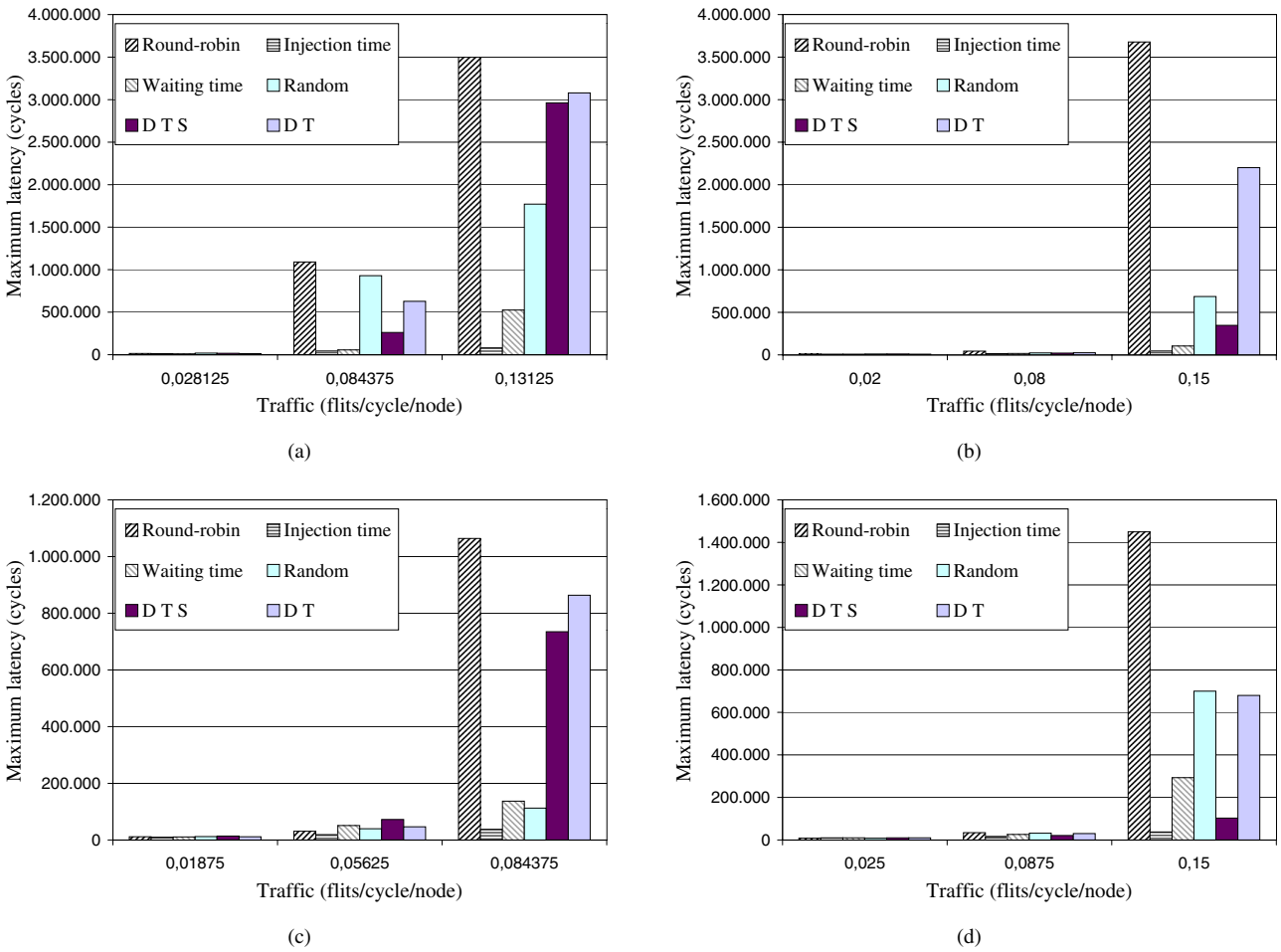

Figure 3. Maximum message latency versus traffic for the destination distributions: (a) uniform, (b) bit-reversal, (c) hot-spot, and (d) local traffic ($l = 3$). Network size is 16 switches and 64 hosts.

throughput by a factor of 1.3 and a reduction in average latency of 40% for medium traffic loads. These policies, though based on different criteria, have in common the purpose of giving priority to those packets that have been in the network for a longer period of time. In general, those packets consume for a long time network resources, and this fact has a negative effect on global network performance (specially in wormhole networks). Therefore, in order to use more efficiently network resources, it is desirable to speed up the forwarding of these packets, as these three policies do.

A new hardware is necessary to implement these arbitration policies. However, the time needed to compute the new necessary information can be overlapped with current timings needed at Myrinet switches and even overlapped with the forwarding of other packets.

As future work we plan to study the effect of the policies with real applications with an execution-driven simulator as well as the impact on different proposed routing algorithms. We also will evaluate the impact of the presented policies in new interconnection networks like InfiniBand.

References

[1] N. Boden, D. Cohen, and R. Felderman. Myrinet – a gigabit per second local area network. *IEEE Micro*, pages 29–36, Feb. 1995.

[2] J. Duato, S. Yalamanchili, and L. Ni. *Interconnection networks. An engineering approach.* IEEE Computer Society, 1997.

[3] J. Flich, M. Malumbres, P. López, and J. Duato. Improving routing performance in myrinet networks. In *Proceedings of International Parallel & Distributed Processing Symposium (IPDPS 2000).*, May 2000.

[4] D. Garcia and W. Watson. Servernet II. In *Lecture Notes in Computer Science*, pages 119–136. Springer, June 1997.

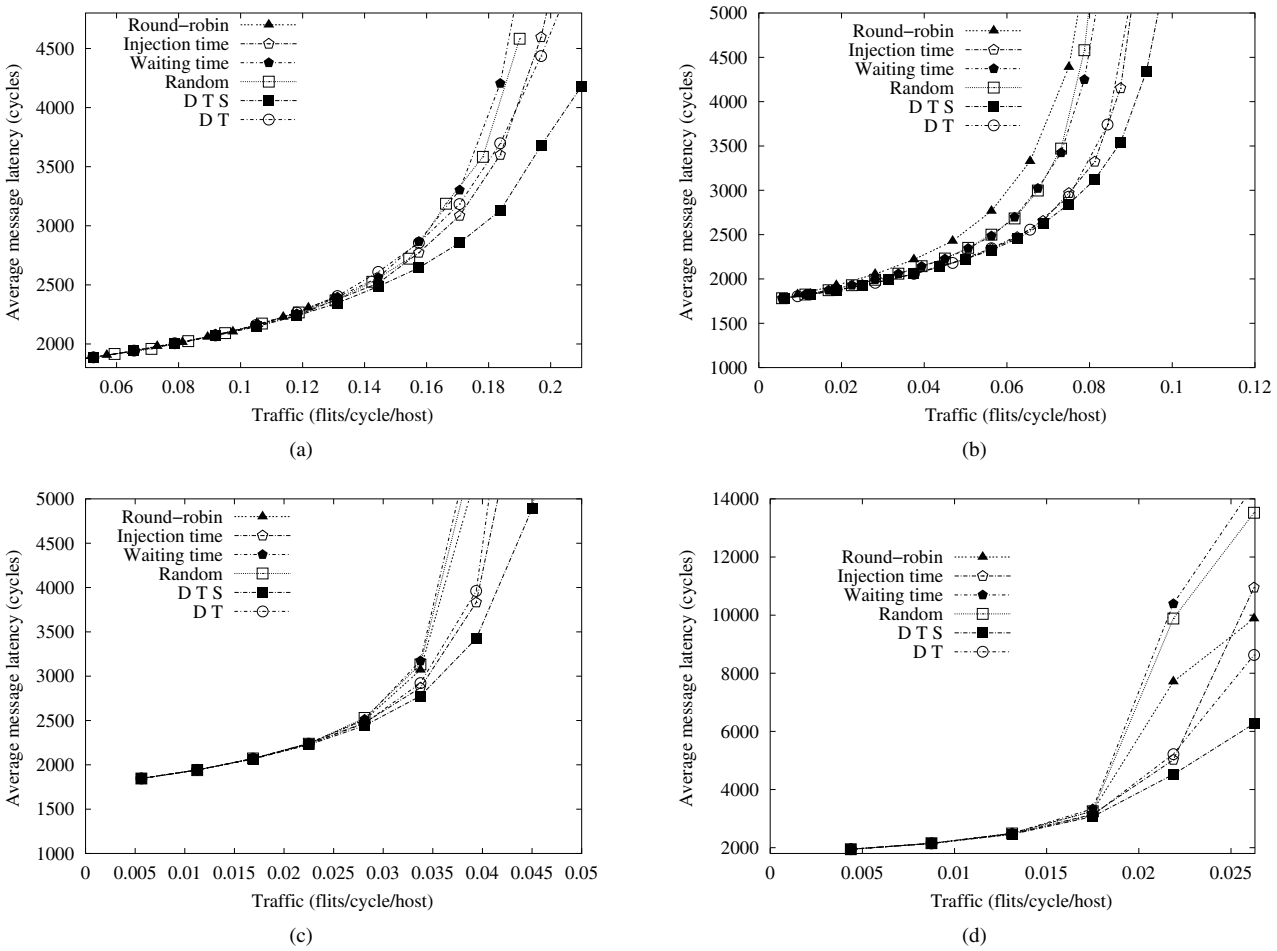

Figure 4. Average message latency versus traffic for the hot-spot message destination distribution, and network size: (a) 8, (b) 16, (c) 32, and (d) 64 switches.

Proceedings of the Workshop on Parallel Computer Routing and Communication.

[5] C. Glass and L. Ni. Adaptive routing in mesh-connected networks. In *Proceedings of the International Conference on Distributed Computing Systems*, pages 12–19, 1992.

[6] R. W. Horst and D. Garcia. Servernet SAN I/O architecture. Technical report, Tandem Computers Incorporated, Aug. 1997.

[7] http://www.myri.com. Myricom home page. Technical report, Myricom, Inc., 2001.

[8] M. Mora. Diseño y evaluación de políticas de selección en los conmutadores de Myrinet. Master's thesis, Dpto. Informática, Universidad de Castilla-La Mancha, 2001. `http://www.info-ab.uclm.es/personal/falfaro/pfc/MDMora.zip`.

[9] M. Pirvu, N. Ni, and L. Bhuyan. Exploring the switch design space in a CC-NUMA multiprocessor environment. In *Proceedings of the 14th International Conference on Parallel and Distributed Processing Symposium (IPDPS-00)*, pages 703–710, Los Alamitos, May 1–5 2000. IEEE.

[10] L. Prylli and B. Tourancheau. BIP: A new protocol designed for high performance networking on Myrinet. *Lecture Notes in Computer Science*, 1388:472–488, 1998.

[11] S. Scott and J. Goodman. The impact of pipelined channels on k-ary n-cubes networks. *IEEE Transactions on Parallel and Distributed Systems*, 5(1):2–16, Jan. 1994.

[12] F. Silla and J. Duato. Improving the efficiency of adaptive routing in networks with irregular topology. In *Proceedings of the 1997 Int. Conference on High Performance Computing*, Dec. 1997.

[13] F. Silla, M. Malumbres, A. Robles, P. López, and J. Duato. Efficient adaptive routing in networks of workstations with irregular topology. In *Proceedings of the Workshop on Communication, Architecture and Applications for Network-based Parallel Computing*, Feb. 1997.